1.) Consider the following decision problem:

**REMOVE VERTEX 3-COL (RV3COL)**

**INSTANCE:** An undirected graph $G = (V, E)$.

**QUESTION:** Does there exists a vertex $x \in V$ such that the removal of $x$ from $G$ yields a 3-colorable graph.

Formally, does there exist $x \in V$ such that the graph $(V \setminus \{x\}, E')$ with $E' = \{(u, v) \in E \mid u \neq x \text{ and } v \neq x\}$ is 3-colorable.

Show NP-hardness of RV3COL by providing a polynomial-time many-one reduction from the standard 3-COL problem. Prove the correctness of your reduction.

Recall that 3-COL is defined as follows:

**3-COL**

**INSTANCE:** An undirected graph $G = (V, E)$.

**QUESTION:** Does there exist a function $\mu$ from vertices in $V$ to values in $\{0, 1, 2\}$ such that $\mu(v_1) \neq \mu(v_2)$ for any edge $[v_1, v_2] \in E$. 
2.) In the exercise part, we considered the Ackermann-Péter function which is as follows.

\begin{algorithm}
\textbf{Algorithm 1}: The Ackermann-Péter function \( \text{AP} \)
\begin{algorithmic}
\Input \( x, y \), two non-negative integers
\Output The computed non-negative integer value for \( x, y \)
\State \textbf{if} \( x == 0 \) \textbf{then}
\State \quad \Return \( y + 1 \);
\State \textbf{else if} \( y == 0 \) \textbf{then}
\State \quad \Return \( \text{AP}(x−1,1) \);
\State \textbf{else return} \( \text{AP}(x−1,\text{AP}(x,y−1)) \);
\end{algorithmic}
\end{algorithm}

Show, using well-founded induction, that

\[ \forall x \forall y \left( (x \in \mathbb{N}_0 \land y \in \mathbb{N}_0) \implies \text{AP}(x,y) > y \right) \]
3.) (a) Let $p$ be the following IMP program:

```
while $y < n$ do
    $x := x + 4 * y + 2$;
    $y := y + 1$
end
```

Prove the total correctness of the following Hoare triple:

$$[n = 10 \land x = 0 \land y = 0] \mathbin{p} [x = 200].$$

(10 points)
(b) Consider the following proof rule:

\[
\begin{array}{c}
\{ A \} \ x := a_1 \ { B } \quad \{ B \land x \leq a_2 \} \ p; \ x := x + 1 \ { B } \\
\hline
\{ A \} \ x := a_1; \textbf{while} \ x \leq a_2 \ \textbf{do} \ p; \ x := x + 1 \ \textbf{od} \ { B } \land x = a_2 + 1
\end{array}
\]

where \( x \) is a variable, \( a_1, a_2 \) are arithmetic expressions, \( p \) is a program, and \( A, B \) are assertions.

Is this proof rule sound? If yes, give a formal proof. Otherwise, give a counterexample.

(5 points)
4.) (a) Provide a non-empty simulation relation $H$ that witnesses $M_1 \leq M_2$, where $M_1$ and $M_2$ are shown below. The initial state of $M_1$ is $s_0$, the initial state of $M_2$ is $t_0$:

**Kripke structure $M_1$:**

- $s_0$: \{a\}
- $s_2$: \{b\}
- $s_3$: \{c\}
- $s_4$: \{a\}
- $s_1$: \{b\}

**Kripke structure $M_2$:**

- $t_0$: \{a\}
- $t_3$: \{b\}
- $t_1$: \{c\}
- $t_4$: \{b\}
- $t_5$: \{c\}
- $t_2$: \{b\}

(4 points)
(b) Consider the following Kripke structure $M$:

For each of the following formulae $\varphi$,

i. check the respective box if the formula is in CTL, LTL, and/or CTL*, and

ii. list the states $s_i$ on which the formula $\varphi$ holds; i.e. for which states $s_i$ do we have $M, s_i \models \varphi$?

\[
\begin{array}{|l|c|c|c|c|}
\hline
\varphi & \text{CTL} & \text{LTL} & \text{CTL*} & \text{States } s_i \\
\hline
G(a) & & & & \\
X(a \land b \land c) & & & & \\
AF(b) & & & & \\
A[(a \land c) U (c)] & & & & \\
E[(b \land c) U (b)] & & & & \\
\hline
\end{array}
\]

(5 points)
(c) LTL tautologies

Prove that the following formulas are tautologies, i.e., they hold for every Kripke structure $M$ and every path $\pi$ in $M$, or find a Kripke structure $M$ and path $\pi$ in $M$, for which the formula does not hold and justify your answer.

i. $Fp \iff ((XG\neg p) \Rightarrow p)$
ii. $(Xp) \ U q \iff X(p \ U q)$

(6 points)